INDUSTRIAL TRAINING REPORT

Internship training report on

Training on UDE, Microcontrollers and its Safety

submitted in partial fulfilment of the
Requirements for the award of

Bachelor of Engineering
in
School of Electronics and Communication Engineering

Carried out at

Bosch Global Software Technologies Private Limited (BGSW)

Submitted By:
Suraj Baddi
01FE19BEC268

Under the guidance of

Prof. Rohit Kalyani	**Mr. Sah Manish Kumar**
College Guide	Industry Guide
KLE Technological University	Bosch Global Software Tech

SUBMITTED TO:
School of Electronics and Communication Engineering
KLE TECHNOLOGICAL UNIVERSITY
Hubballi

K.L.E SOCIETY'S
KLE Technological University,
HUBBALLI-580031
2022-2023

SCHOOL OF ELECTRONICS AND COMMUNICATION ENGINEERING

CERTIFICATE

This is to attest to the fact that the student **" Suraj Baddi [01FE19BEC268] "** studying in final year has undergone Industrial internship training at **Bosch Global Software Technologies** in Bangalore from **20/01/2023 to 31/05/2023** in partial fulfillment for the award for Bachelor of Engineering in Electronics and Communication in the School of Electronics and Communication Engineering of KLE Technological University, Hubballi for the academic year 2022-2023.

Prof. Rohit Kalyani	Dr. Nalini C. Iyer	Dr. Basavaraj S. Anami
Guide	Head of School	Registrar

External Viva:

Name of Examiners **Signature with date**

1.

2.

ACKNOWLEDGMENT

The happiness and exhilaration that accompany the successful completion of any task would be insufficient if I did not acknowledge a number of people whose expert advice and encouragement assisted me in the successful completion of this report work. I take this opportunity to thank Dr. Ashok Shettar, Vice-chancellor, KLE Technological University and to Dr. Prakash Tewari, Principal, KLE Technological University, Hubballi. I also take this opportunity to thank Dr. Nalini Iyer, Head of School, School of Electronics and Communication Engineering for providing the opportunity of industry internship at Bosch Global Software Technologies which nurtured our practical skills contributing to the success of our project.

I sincerely thank my guide Mr. Sah Manish Kumar (MS/EEH21-PS), Bosch Global Software Technologies Private Limited (BGSW) for his guidance, inspiration and wholehearted cooperation during the course of completion. I sincerely thank Mr. Rudrappa Gouda (MS/EEH22-PS) and Mr. Prakash Shetty (MS/EEH2-PS), for the valuable guidance and advice who inspired me greatly to work in this project. Their willingness to motivate, contributed tremendously to the project.

I would also like to thank Prof. Rohit Kalyani, School of Electronics and Communication Engineering for his support and encouragement.

- Suraj Baddi (01FE19BEC268)

ABSTRACT

The rapid advancements in technology have led to the widespread use of microcontrollers in various industries, ranging from automotive to consumer electronics. However, ensuring the safety and reliability of microcontroller-based systems is of paramount importance to prevent potential hazards and malfunctions. This training program aims to equip participants with the necessary knowledge and skills to effectively utilize UDE, a versatile development environment for microcontrollers, while prioritizing safety considerations. The training covers key topics such as microcontroller architecture, programming techniques, debugging methodologies, and safety standards applicable to microcontroller systems. Participants will gain hands-on experience through practical exercises and case studies, enabling them to identify potential safety issues, implement appropriate safety measures, and validate the safety of microcontroller-based systems. By attending this training, professionals and enthusiasts will enhance their expertise in working with microcontrollers and contribute to the development of safe and reliable embedded systems across various industries.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In today's technology-driven world, microcontrollers play a crucial role in numerous industries, ranging from automotive and aerospace to consumer electronics and industrial automation. These small, embedded systems provide the backbone for powering various devices and applications, making them an integral part of modern technology. As the complexity and functionality of microcontrollers continue to evolve, so does the need for effective debugging and safety measures to ensure reliable and secure operation.

The Universal Debug Engine (UDE) stands as a powerful tool in the realm of microcontroller development, offering comprehensive debugging capabilities for a wide range of microcontroller architectures. Developed by PLS Programmierbare Logik and Systeme GmbH, UDE supports various microcontroller families, including those from leading manufacturers such as Infineon and STMicroelectronics (ST). By enabling efficient debugging, UDE significantly aids developers in identifying and rectifying software and hardware issues, thus enhancing the overall quality and reliability of microcontroller-based systems.

This paper focuses on exploring the training aspects of UDE, microcontrollers from Infineon and ST, and the safety considerations associated with their use. We delve into the essential concepts and techniques involved in training developers on UDE and the intricacies of microcontroller development for Infineon and ST platforms. Furthermore, we investigate the safety challenges and measures necessary to ensure the robust operation of microcontroller-based systems, considering both functional safety and security aspects.

The goals of this paper are twofold. Firstly, we aim to provide a comprehensive overview of UDE, its features, and its significance in the field of microcontroller development. We discuss the debugging capabilities it offers, its compatibility with Infineon and ST microcontrollers, and the benefits it brings to the development process. Secondly, we address the critical aspect of safety, emphasizing the importance of adhering to safety standards and best practices when designing and deploying microcontroller-based systems. We explore the specific safety considerations associated with Infineon and ST microcontrollers, shedding light on the challenges faced and the strategies employed to ensure safe and reliable operation.

1.1 Company Background

Bosch Global Software Technologies (BGSW) is the software development unit of Bosch, a leading global supplier of technology and services. BGSW was established in 2011 with the aim of developing cutting-edge software solutions that enable Bosch to stay ahead of the curve in the rapidly evolving technology landscape. With a workforce of over 15,000 associates spread across 30 countries, operates in a wide range of industries such as automotive, healthcare, energy, and building technology.

Figure 1.1: Kormangala Bosch

BGSW's software development expertise encompasses a wide range of technologies, including artificial intelligence, machine learning, cloud computing, and the Internet of Things (IoT). The company's software solutions cover a broad spectrum of applications, from automotive and industrial automation to smart homes and buildings, and from healthcare and energy management to connected mobility.

BGSW's commitment to quality is reflected in its adherence to international software development standards such as Automotive SPICE, ISO 26262, and IEC 61508. The company also places a strong emphasis on data privacy and security, ensuring that its software solutions are designed to meet the highest standards of cybersecurity.

1.2 Training Objectives

The following are the key objectives which were focused:

1. Understanding the Need for Functional Safety and the ISO26262 standard.
2. To study the basics of different microcontrollers (Infineon and STmicroelectronics)
3. To research the safety aspects particular to these controllers.
4. To learn about UDE (Universal Debug Engine), a tool used for system analysis, testing, and debugging.

Chapter 2

Functional Safety: ISO 26262

In this Section, we will look onto the basic concepts of Functional Safety, its lifecycle and the analysis techniques.

2.1 Definition of Functional Safety

The "Road Vehicles - Functional Safety" standard ISO 26262 provides guidelines for guaranteeing the functional safety of electrical and electronic systems in motor vehicles. It outlines a safety lifecycle that encompasses the entire development process and a risk-based methodology. It is frequently necessary for suppliers of safety-critical systems and components to car OEMs to comply with the standard.

Functional safety is characterised by ISO 26262 as the "absence of unreasonable risk caused by hazards resulting from the malfunctioning behaviour of Electrical/Electronic systems." This contractual concept can be visualised as a series of consequences, where a failure of an E/E component causes a hazard to materialise, which in turn creates a threat of harm or damage. As a result, a risk level that is deemed acceptable is used to establish the necessary risk reduction strategies.

2.2 Failure Classification

The ISO 26262 standard divides failures of the E/E component into two categories:

Systematic failures:

Systematic failures are those that are caused by issues in the development, manufacturing, or maintenance processes of a product or system. They are often induced in a deterministic or predictable way and affect a large number of items or functions.

Random failures:

These failures are caused by random defects that are innate to the process or usage conditions and can be difficult to predict or prevent. To mitigate the risk of hardware random failures, hardware designers and manufacturers often use redundancy, error correction codes, and other techniques to improve the reliability and resilience of their products.

More specifically, there are two types of hardware random failures: those that are temporary (like single-event faults or soft mistakes) and those that are permanent (like stuck-at faults). The HW/SW system's design and verification process addresses handling random failures by including safety measures that enable the architecture to recognise and fix errors. According to

ISO26262, safety mechanisms are technological ways to identify and address problems in order to maintain a safe condition. A fault is often described as an anomalous circumstance that may result in failure. Examples of safety techniques include ECC (Error Correction Code), CRC (Cyclic Redundant Check), hardware redundancy, BIST (Built-In Self-Test), and a number of other processes.

2.3 ASIL (Automotive Safety Integrity Level)

After a specified vehicle component (such the ABS anti-lock braking system) fails, a hazard and risk analysis is conducted to evaluate the potential for human injuries and property damage. The ASIL (Automotive Safety Integrity Level), or the amount of risk reduction required to achieve a tolerable risk, is determined by this study, which is based on the Exposure, Severity, and Controllability of the hazard and the consequent risk. ASIL A is the level of safety reduction that is the least severe, while ASIL D is the most significant. Figure 2.1 provides an example of the steps leading to the ASIL conclusion based on the fault and its potential impact.

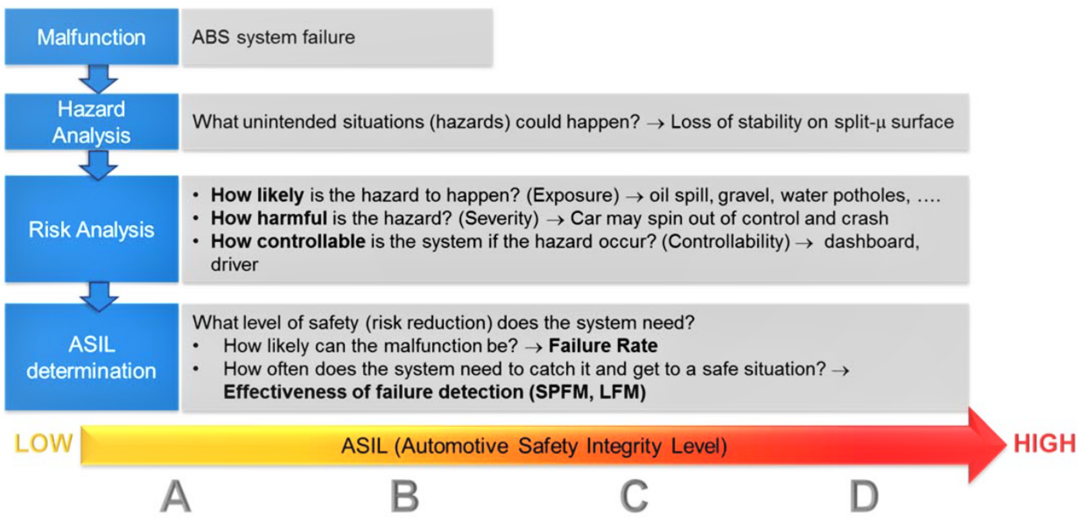

Figure 2.1: ABS example of ASIL determination based on Hazard and Risk Analysis.

2.4 Functional Safety Lifecycle and Development Phases

The ISO 26262 standard provides a detailed definition and documentation of the functional safety lifecycles. While Figure 2.2a depicts the concept and development phases, Figures 2.2b and 2.2c provide examples of the respective Functional Safety activities. In the idea phase, which is controlled by manufacturers, the systems that will be used to implement a function at the vehicle level (referred to as a "item" in ISO26262 language), such as the Automatic Emergency Braking System, are described. At this level, the ASIL is acknowledged, and safety objectives and functional safety standards are developed in accordance with it. When the system level product development phase begins, the technical safety requirements are derived with respect to the HW/SW components of the safety-related function for each functional safety need.

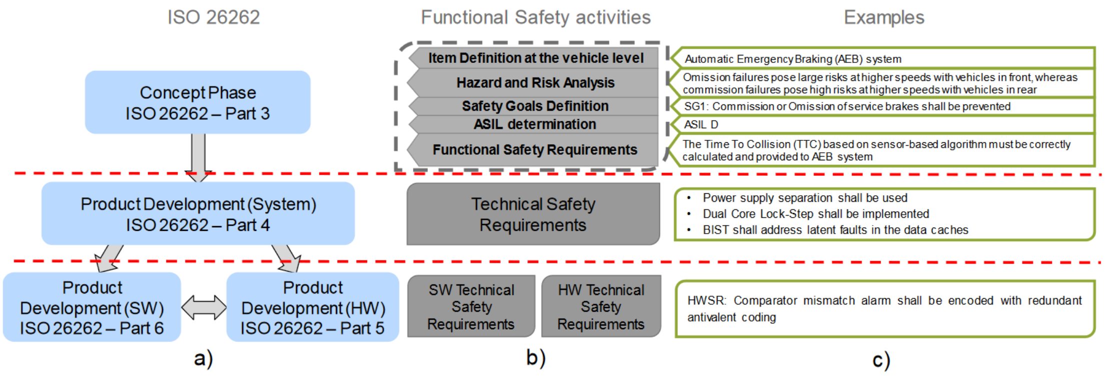

Figure 2.2: Phases of the Functional Safety development process, corresponding requirements and examples

2.5 Timing Analysis

The evaluation of safety systems must consider timing performance as a crucial factor. The system's ability to detect faults and transition to a safe state within a defined time, known as the Fault Tolerant Time Interval (FTTI), is essential to prevent potential system-level hazards. Figure 2.3 illustrates this concept, including the Diagnostic Test Interval (DTI) allocated for defect detection. For instance, a CPU's fault detection DTI may be around 10ms, while the system's FTTI would exceed 100ms. ASIL B requires 60% diagnostic coverage for latent faults and 90% for single point faults, while ASIL D requires 90% for latent faults and 100% for single point faults. Safety measures like Error Correcting Code (ECC) in embedded memories can detect single point faults, identifying up to two-bit errors during memory read operations. Hardware Built-In Self-Test (BIST) serves as a latent fault detection tool to find faults in ECC encode and decode logics, enhancing system integrity.

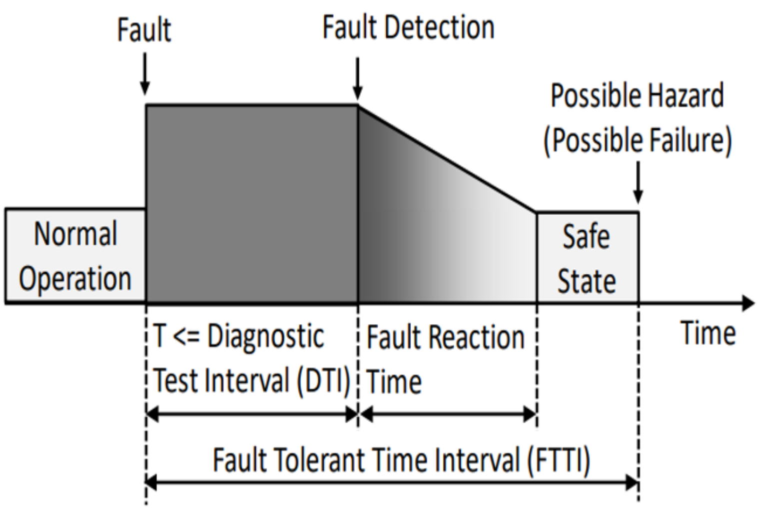

Figure 2.3: Fault Tolerant Time Interval and Diagnostic Test Interval

Chapter 3

Basics of Microcontrollers and its Safety

In this chapter, we'll go over the fundamentals of the Infineon device 4 and STmicroelectronics SPC58 Bernina microcontrollers, as well as their safety features and failure reactions.

3.1 Infineon Device 4

Infineon Device 4 is a microcontroller from Infineon Technologies, designed for a wide range of applications, particularly in the automotive industry. It belongs to the AURIX™ family of microcontrollers known for their high performance, safety, and security features.

Figure 3.1: Infineon Controller

3.1.1 Features of IFX device 4

Here are some key aspects of Infineon Device 4:

1. **Core Architecture:** Infineon Device 4 is based on the ARM Cortex-R4F core, offering a balance between processing power and real-time performance. It is capable of handling complex automotive applications that require real-time responsiveness.

2. **Safety and Security:** Infineon Device 4 is designed with a strong focus on functional safety and security. It complies with the ISO 26262 standard, making it suitable for safety-critical automotive systems. It incorporates features like hardware safety mechanisms, integrated self-test capabilities, and advanced security features to protect against cyber threats.

3. **Peripherals and Connectivity:** Infineon Device 4 offers a rich set of integrated peripherals, including CAN, LIN, SPI, I2C, and Ethernet interfaces, enabling seamless communication with other automotive systems and components. It also includes advanced control peripherals for motor control applications.

4. **Memory:** Infineon Device 4 provides a range of memory options, including flash memory for program storage and various types of RAM for data storage. This allows developers to choose the appropriate memory size for their specific application requirements.

5. **Automotive-Specific Features:** Infineon Device 4 incorporates automotive-specific features to meet industry requirements. These features include support for automotive communication protocols (e.g., CAN, LIN), motor control interfaces, and integrated self-test capabilities to ensure system reliability and fault detection.

6. **Power Management:** Infineon Device 4 includes power management features to optimize energy consumption in automotive applications. It offers power-saving modes and supports efficient power management techniques to enhance overall system efficiency.

7. **Automotive-Specific Features:** Infineon Device 4 incorporates automotive-specific features to meet industry requirements. These features include support for automotive communication protocols (e.g., CAN, LIN), motor control interfaces, and integrated self-test capabilities to ensure system reliability and fault detection.

3.1.2 Applications of IFX device 4

Infineon Device 4, being a member of the AURIX™ family of microcontrollers, is primarily targeted towards automotive applications that require high performance, safety, and security. Some specific applications of Infineon Device 4 include:

1. **Advanced Driver Assistance Systems (ADAS):** Infineon Device 4 is used in ADAS applications such as adaptive cruise control, lane departure warning, and collision avoidance systems. Its real-time capabilities and safety features make it suitable for critical functions in autonomous and semi-autonomous vehicles.

2. **Electric Powertrain Systems:** With the growing popularity of electric vehicles (EVs) and hybrid electric vehicles (HEVs), Infineon Device 4 is utilized in powertrain systems for motor control, battery management, and energy conversion. Its high performance and safety features ensure efficient and reliable operation of electric powertrain components.

3. **Chassis and Safety Systems:** Infineon Device 4 is employed in various chassis and safety systems, including anti-lock braking systems (ABS), electronic stability control (ESC), and airbag control units. Its real-time responsiveness and safety mechanisms help enhance vehicle safety and stability.

4. **Body Electronics:** In automotive body electronics applications, Infineon Device 4 is utilized for functions like central body control, lighting control, and climate control. Its integrated peripherals and connectivity options enable seamless communication and control of various vehicle systems.

3.2 STmicroelectronics SPC58 (Bernina)

ST SPC58 is a family of microcontrollers from STMicroelectronics, designed for automotive and industrial applications that demand high performance, safety, and reliability.

Figure 3.2: STmicroelectronics controller

3.2.1 Features of Bernina

1. **Core Architecture:** ST SPC58 microcontrollers are based on the Power Architecture® core, providing high processing power and efficient execution of complex tasks in automotive and industrial applications. The core offers excellent performance and real-time capabilities.

2. **Safety and Security:** ST SPC58 microcontrollers prioritize functional safety and security. They are designed to comply with the ISO 26262 standard, making them suitable for safety-critical applications. The microcontrollers integrate safety mechanisms, error correction codes, and memory protection units to ensure reliable and secure operation.

3. **Peripherals and Connectivity:** ST SPC58 microcontrollers offer a wide range of integrated peripherals, enabling seamless communication and control of various automotive and industrial systems. These peripherals include CAN, LIN, SPI, I2C, Ethernet, and USB interfaces. They also provide advanced control peripherals specifically designed for motor control applications.

4. **Memory Options:** ST SPC58 microcontrollers offer flexible memory configurations to accommodate program storage and data storage requirements. They include flash memory for program storage and different types of RAM for data storage. The memory size options vary across different microcontroller models, allowing developers to choose the right configuration for their applications.

5. **Automotive-Specific Features:** ST SPC58 microcontrollers incorporate automotive-specific features to meet industry requirements. They support automotive communication protocols (such as CAN, LIN) and include features for motor control, automotive networking, and system monitoring. These features enable seamless integration into automotive applications.

3.2.2 Applications of Bernina

ST SPC58 microcontrollers are designed for automotive and industrial applications, providing high performance, safety, and reliability. Some common applications of ST SPC58 microcontrollers include:

1. **Engine Management Systems:** ST SPC58 microcontrollers are utilized in engine control units (ECUs) for managing various aspects of the engine, including fuel injection, ignition timing, and emissions control. Their processing power and real-time capabilities enable precise control and optimization of engine performance.

2. **Advanced Driver Assistance Systems (ADAS):** ST SPC58 microcontrollers play a crucial role in ADAS applications such as adaptive cruise control, lane departure warning, and collision avoidance systems. Their high-performance computing capabilities and integrated peripherals facilitate real-time data processing and decision-making for enhanced vehicle safety.

3. **Body Control Modules (BCM):** ST SPC58 microcontrollers are employed in BCMs, which control and monitor various functions related to the vehicle body. These functions include lighting control, climate control, power windows, central locking, and more. The microcontrollers' robust connectivity options and integrated peripherals enable efficient communication and control of body-related systems.

4. **Infotainment Systems:** ST SPC58 microcontrollers are used in automotive infotainment systems to enable multimedia functionality, connectivity features, and user interface control. They provide processing power and interfaces necessary for multimedia playback, navigation systems, connectivity with external devices, and touch screen control.

5. **Electric Powertrain Systems:** With the rise of electric vehicles (EVs) and hybrid electric vehicles (HEVs), ST SPC58 microcontrollers are employed in powertrain systems for motor control, battery management, and energy conversion. Their advanced control peripherals and safety features ensure efficient and reliable operation of electric powertrain components.

6. **Safety and Chassis Systems:** ST SPC58 microcontrollers are utilized in safety-critical systems such as anti-lock braking systems (ABS), electronic stability control (ESC), and airbag control units. Their real-time capabilities, safety mechanisms, and connectivity options contribute to enhanced vehicle safety and stability.

7. **Instrument Clusters:** ST SPC58 microcontrollers are integrated into instrument clusters to provide visual and graphical information to the driver. They handle the processing and display of critical vehicle information such as speed, fuel level, engine status, and warnings. The microcontrollers' performance and display interface support enable efficient and intuitive information presentation.

3.3 Safety Modules

Safety modules in a microcontroller are dedicated hardware components or features designed to enhance the safety and reliability of the system. These modules typically incorporate various safety mechanisms and functionalities to detect and mitigate faults, ensuring the system operates within defined safety limits.

3.3.1 Infineon Safety Modules

When hardware detects any errors, it activates an alarm that is directed to the SMU Error Management Module (EMM). Similarly, software can also trigger alarms. If errors are detected by the ECC decoders, the associated SRAM's MBIST logic signals the EMM. The SMU EMM receives and consolidates over-voltage and under-voltage status signals, generating the appropriate EVR (Error Voltage Regulation) status. In accordance with the safety concept, the SMU collects all alarm flags from various hardware monitors. This ensures comprehensive error monitoring and management within the microcontroller system.

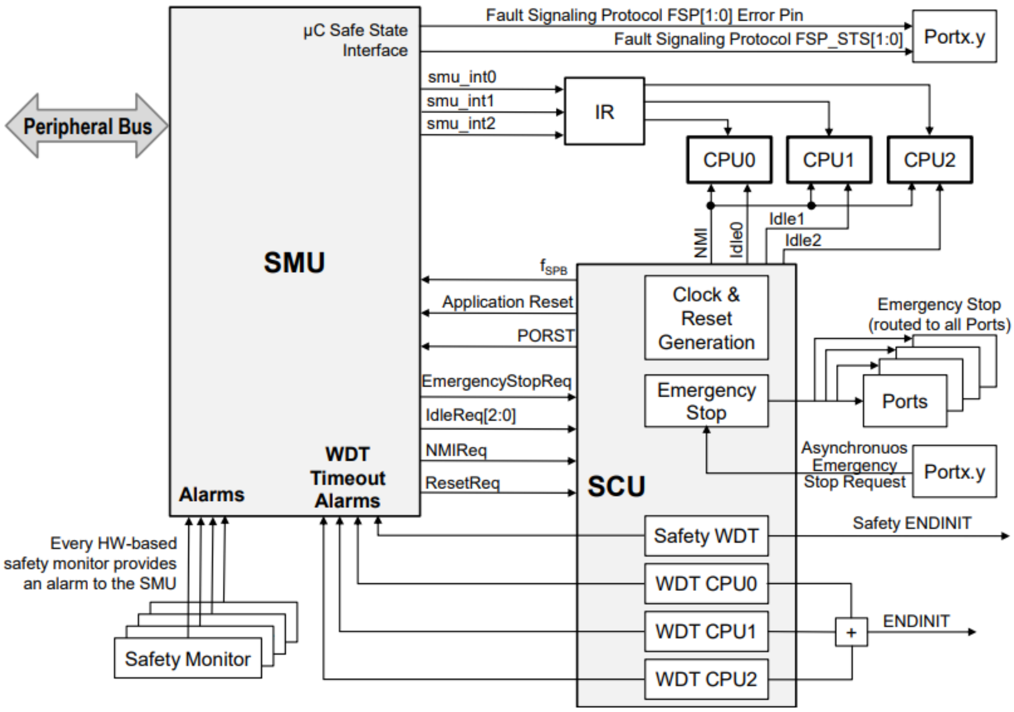

Figure 3.3: Overview of Infineon Safety modules

3.3.2 ST Safety Modules

The microcontroller device supports a Fault Tolerant Time Interval (FTTI) of at least 10 ms. Hardware-detected faults are reported to the central Fault Collection and Control Unit (FCCU) for monitoring. The FCCU utilizes redundancy checkers, self-test features, Error Correction Code (ECC), voltage monitors, and clock monitors to ensure fault detection. In the event of a fault, the FCCU configures the device into the appropriate fail-safe state, including shutdown or reset. Internal faults are indicated to the surrounding system through dedicated Error Out signals. Safety critical I/Os are tri-stated when the device enters the fail-safe state. The frequency meter in CMU continuously monitors the frequency of the Internal RC (IRC) clock. It is important to note that the failure of the IRC clock is considered a catastrophic failure since it serves as the boot and backup clock for the device. The FCCU does not automatically trigger an error condition in the event of IRC failure.

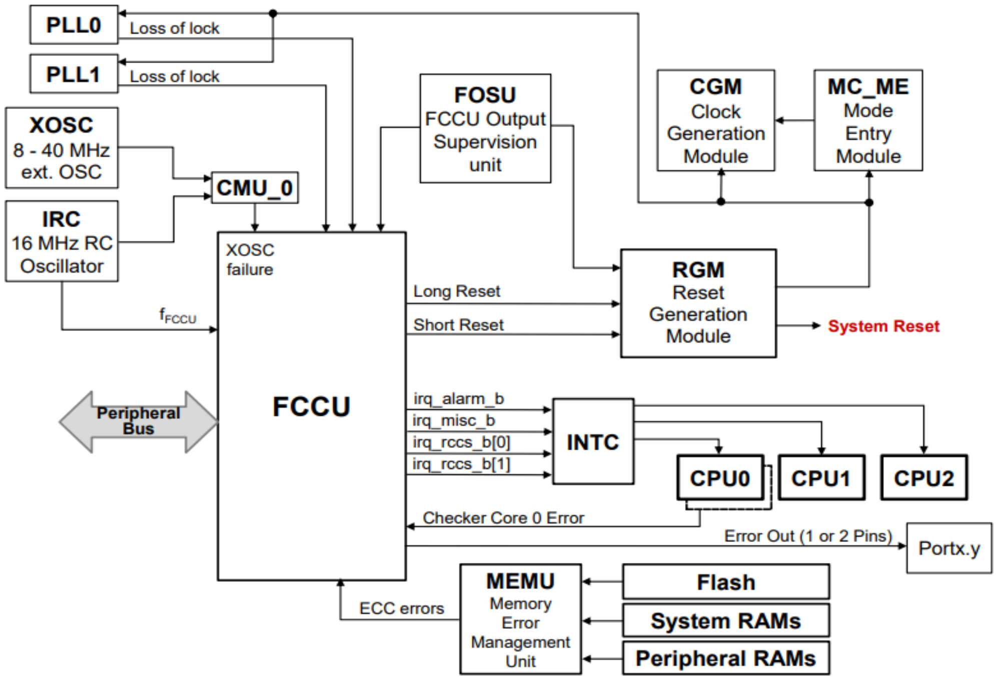

Figure 3.4: Overview of ST Safety modules

3.4 Comparision between device Safety features

Table 3.1: Device Safety Features

	Infineon Device	**ST Device**
Safety Core (Core with Lockstep)	Computational Core	Computational Core
Memory Monitoring	Memory Test Unit (MTU)	Memory Error Management Unit (MEMU)
Bus Transfers	E2E ECC	E2E ECC
CPU Watchdog Timer	• one WDT for each core. • one safety WDT.	• one SWT for each CPU. • one security SWT.
Clock Monitors	• Internal oscillator clock. • PLL-generated clock. • FlexRay clock.	• Internal oscillator clock. • PLL-generated clock.
Voltage Monitors	• Low voltage detector. • High voltage detector.	• Low voltage detector. • High voltage detector.

Chapter 4

Software and Hardware

In this chapter, we will look at the software and hardware that will be used for testing debugging and system analysis.

4.1 UDE (Universal Debug Engine)

The UDE (Universal Debug Engine) is an advanced tool designed for debugging, testing, and system analysis purposes. Offering a wide array of features, the UDE seamlessly integrates comprehensive capabilities for debugging, trace analysis, and runtime analysis, while maintaining an intuitive and efficient user interface. Key functionalities of the UDE include debugging support for C/C++ and assembler languages, real-time monitoring, system visualization, and system analysis. Additionally, the UDE extends its compatibility to encompass a diverse range of multicore System-on-Chips (SoCs) and microcontroller families.

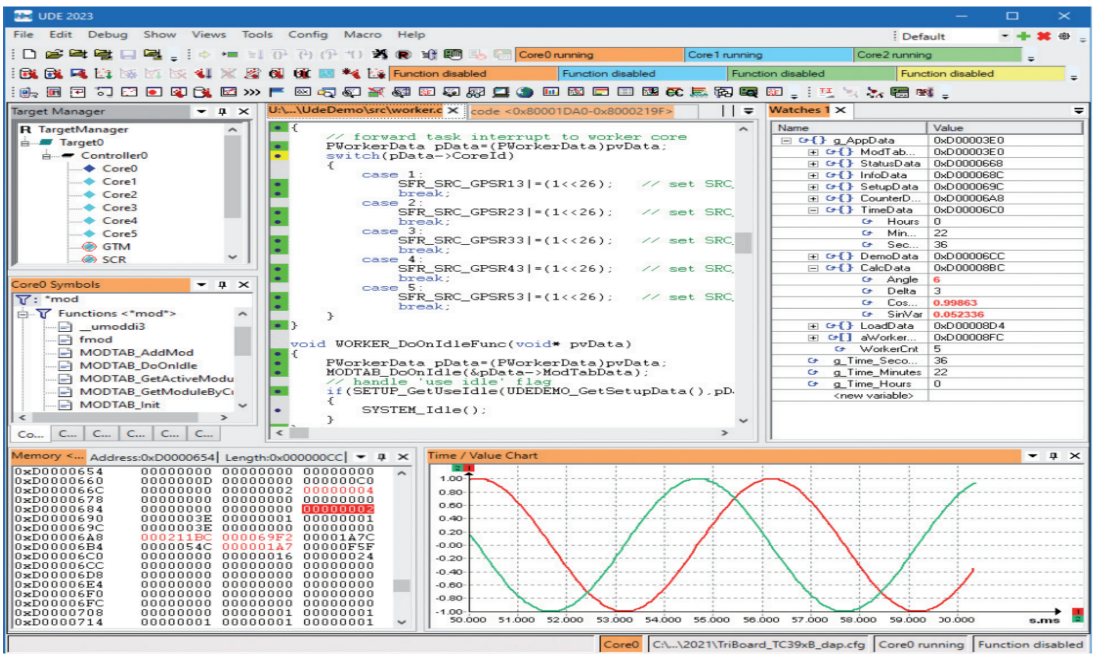

Figure 4.1: Glance of UDE Software

UDE provides various features and functionalities to aid developers in identifying and resolving software bugs and issues, here are some common features you might find in a Universal Debug Engine:

1. **Target Connection:** UDE allows you to establish a connection between the debugging tool and the target system, typically through a debugging interface like JTAG or SWD. It enables communication and control between the host computer and the target microcontroller.

2. **Breakpoints and Watchpoints:** UDE supports setting breakpoints, which allow you to pause the execution of the program at specific lines of code or memory addresses. Watchpoints, on the other hand, allow you to monitor changes in specific variables or memory locations and halt the program when those changes occur.

3. **Stepping and Flow Control:** UDE provides various stepping options, such as single-step, step over, and step out, to control the flow of program execution. This allows you to navigate through the code and examine its behavior in detail.

4. **Variable and Memory Inspection:** UDE enables you to inspect the values of variables and memory locations during runtime. You can view and modify variable values, examine memory contents, and track their changes as the program executes.

5. **Real-time Trace:** Some versions of UDE may support real-time trace capabilities, which allow you to capture and analyze the execution flow of your program in detail. This can be useful for understanding timing issues, performance optimization, and system-level debugging.

6. **Peripheral and Register Access:** UDE provides access to the microcontroller's peripherals and registers, allowing you to read and modify their values directly. This feature is particularly useful for low-level hardware debugging.

7. **Multicore Debugging:** For systems with multiple cores or processors, UDE may offer support for multicore debugging. It allows you to debug each core independently or synchronize their execution for debugging purposes.

8. **Scripting and Automation:** UDE often provides scripting capabilities, allowing you to automate repetitive debugging tasks or customize the debugging environment to your specific needs. Scripting languages like Python or JavaScript are commonly supported.

9. **Flash Programming:** UDE may include features for programming the microcontroller's flash memory with the compiled firmware or software updates.

4.2 JTAG (Joint Test Action Group)

JTAG (Joint Test Action Group) is a standardized interface used for testing, debugging, and programming integrated circuits. It allows for debugging access, boundary scan testing, in-circuit emulation, and programming of devices. It provides a standardized connection and control mechanism for efficient and reliable testing and debugging of electronic systems.

4.3 UAD (Universal Access Device) 2 pro

The UAD2pro is a high-performance universal access device that provides debugging and trace capabilities for various microcontrollers and processors from different manufacturers. It serves as a bridge between a host computer and the target system, enabling software developers to analyze, debug, and optimize the code running on the target system. It offers fast communication with the target system, allowing for efficient debugging and analysis. The UAD2pro may support different communication interfaces such as JTAG, cJTAG, DAP, SWD, or others, depending on the specific version.

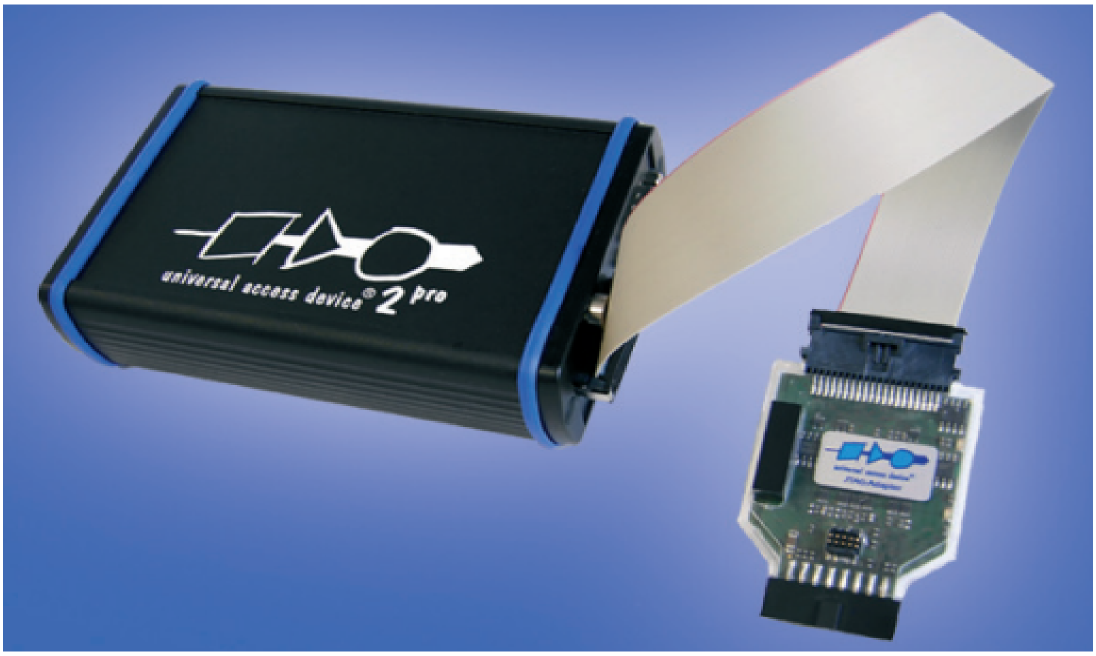

Figure 4.2: Glance of UDE Software

4.4 Infenion Device 4 (TC389QP)

The second-generation AURIX microcontroller from Infineon is designed for automotive applications. It offers high performance with 6 TriCore cores running at 300 MHz, up to 16 MB flash memory, and 6.9 MB SRAM. It provides extensive connectivity options, robust security features, and complies with safety standards up to ASIL-D. The microcontroller supports over-the-air software updates, scalability, and easy migration from previous AURIX™ versions.

4.5 Hardware Integration

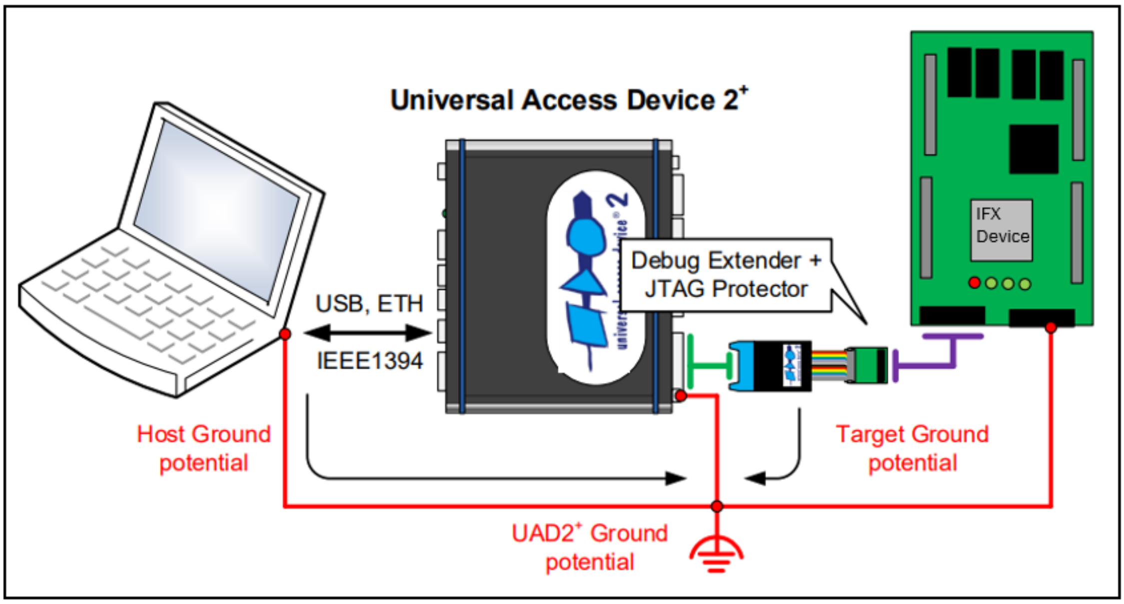

Figure 4.3: H/W Integration

Chapter 5

Conclusion and Learnings

Being a part of Bosch internship program was an excellent experience for me since I learned a lot of technical knowledge and gained practical hands-on microcontrollers used in automotive sectors. I enjoy how interns are treated like full-time workers, with real and meaningful duties that are important and valuable to Bosch services. I obtained a better understanding of the Automotive sector, which has aided me much in my career plans. Through the internship, I gained a greater understanding of the professional path of a Automotive engineer, which assisted me in making an educated career decision. The new long-term ties and connections I had built with my Bosch colleagues and fellow interns were the most important takeaways for me.

I am grateful to have been given this opportunity since it has allowed me to grow as a person through the different obstacles that have been offered to me.
My key takeaways were:

- **Be pro-active** - Look for work so that you will have the chance to learn as well as to contribute.
- **Interact** - Make work buddies and network; you never know when an opportunity will present itself.
- **Be yourself** - Internships are a time for employers to get to know you, as well as for you to get to know your potential employer and determine whether the two of you are a good match; be yourself so that you can make an educated decision at the end of the day.

train

ORIGINALITY REPORT

8%	8%	%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.coursehero.com Internet Source	3%
2	www-blue.infineon.com Internet Source	1%
3	utpedia.utp.edu.my Internet Source	1%
4	jfitzsimons.com Internet Source	<1%
5	kabar.unismuhpalu.ac.id Internet Source	<1%
6	www.kscst.iisc.ernet.in Internet Source	<1%
7	www.st.com Internet Source	<1%
8	onlinelibrary.wiley.com Internet Source	<1%
9	www.koreascience.or.kr Internet Source	<1%

10	www.mdpi.com Internet Source	<1%
11	scholarworks.uark.edu Internet Source	<1%
12	brainonboard.ca Internet Source	<1%
13	www.researchgate.net Internet Source	<1%
14	www.studymode.com Internet Source	<1%
15	doi.org Internet Source	<1%
16	dokumen.pub Internet Source	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		